Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (*Previously presented*) A cache controller for use with a processor, comprising:

a plurality of mappers for receiving instructions of an instruction set, each mapper for mapping an instruction of said instruction set to a predetermined instruction width format (PIWF) configuration, wherein said plurality of mappers include

at least one first mapper for receiving instructions from a fill buffer, and

at least one second mapper for receiving instructions from an instruction cache;

a multiplexor for receiving said PIWF configurations from said plurality of mappers and selecting, in response to a selector signal, a desired one of said PIWF configurations for decoding and execution by the processor; and

a decoder for decoding the desired one of said PIWF configurations for execution by the processor.

2. (*Original*) The cache controller of claim 1, further comprising:

a tag comparator for generating said selector signal.

3. (*Previously presented*) The cache controller of claim 2, wherein said tag comparator comprises:

a comparer that compares each instruction provided to one of said plurality of mappers with a tag associated with an instruction of said instruction set to a desired tag and generate said selector signal to cause said multiplexor to select said desired one of said PIWF configurations based on the comparison.

4. (*Cancelled*)

5. (*Previously presented*) A method for mapping an instruction set to a predetermined instruction width format (PIWF) configuration, comprising:

(a) reading instructions of said instruction set from an instruction cache and a fill buffer into a plurality of mappers, wherein at least one of said instructions is read from said instruction cache and at least one of said instructions is read from said fill buffer, each instruction of said instruction set being read into a corresponding one of said plurality of mappers in preparation for mapping;

(b) mapping each instruction of said instruction set to a corresponding PIWF configuration;

(c) selecting a desired one of said corresponding PIWF configurations, after said mapping, for decoding and execution by the processor; and

(d) decoding the desired one of said PIWF configurations for execution by the processor.

6. (*Previously presented*) The method of claim 5, further comprising:

(e) comparing, for each instruction provided to one of said plurality of mappers, a tag associated with an instruction of said instruction set to a desired tag, wherein said desired one of said PIWF configurations is selected based on said comparison.

7. (*Cancelled*)

8. (*Previously presented*) A processor comprising:

an execution unit;

a decoder;

a cache for storing instructions; and

a cache controller for retrieving said instructions from said cache and providing said instructions to said decoder, said cache controller comprising:

a plurality of mappers for mapping a plurality of instructions of an instruction set to predetermined instruction width format (PIWF) configurations said plurality of mappers including at least one first mapper for receiving instructions from a fill buffer, and at least one second mapper for receiving instructions from said instruction cache,

a multiplexor for selecting and receiving, in response to a selector signal, one of said PIWF configurations from said plurality of mappers for decoding by said decoder and execution by said execution unit, and

a comparer that compares each instruction provided to said multiplexor with a tag associated with an instruction of said instruction set to a desired tag and generate said selector signal to cause said multiplexor to select said desired one of said PIWF configurations,

whereby said processor performs instruction mapping substantially in parallel with tag comparison to improve processor performance, and

whereby said decoder decodes the desired one of said PIWF configurations for execution by the processor based on the comparison.

9. (*Cancelled*)

10. (*Currently amended*) A computer ~~program product comprising a computer~~ readable storage medium~~, the computer readable storage medium~~ having embodied thereon computer readable program code for providing a microprocessor core, including a cache controller, the computer readable program code comprising:

first computer readable program code for providing a plurality of mappers for receiving instructions of an instruction set, each mapper for mapping an instruction of said instruction set to a predetermined instruction width format (PIWF) configuration, wherein said plurality of mappers include

at least one first mapper for receiving instructions from a fill buffer, and

at least one second mapper for receiving instructions from an instruction cache;

second computer readable program code for providing a multiplexor for receiving said PIWF configurations from said plurality of mappers and selecting, in response to a selector signal, a desired one of said PIWF configurations for decoding and execution by said microprocessor core, and

third computer readable program code for decoding the desired one of said PIWF configurations for execution by the processor.

11. (*Previously presented*) The computer program product of claim 10, further comprising:

fourth computer readable program code for providing a tag comparator, configured to compare, for each instruction provided to one of said plurality of mappers, a tag associated with an instruction of said instruction set to a desired tag and to generate said selector signal to cause said multiplexor to select said desired one of said PIWF configurations.

12. (*Previously presented*) The computer program product of claim 10, wherein the computer readable program code is embodied in hardware description language software.

13. (*Previously presented*) The computer program product of claim 12, wherein the computer readable program code is embodied in Verilog hardware description language software.

14. (*Previously presented*) The computer program product of claim 12, wherein the computer readable program code is embodied in VHDL hardware description language software.

15-20. (*Cancelled*)

21. (*Currently amended*) A method for decoding instructions in a processor, comprising:

(a) mapping each of a plurality of instructions to a predetermined instruction width format (PIWF) configuration;

(b) comparing, in parallel with (a), a tag for each of said plurality of instructions to an address, wherein each ~~tag is associated with a single~~ instruction is associated with a unique tag;

(c) selecting, based on the comparison in (b), one of the PIWF configurations of (a) to be decoded; and

(d) decoding the PIWF configuration selected in (c) for execution by a processor core.

22. (*Previously Presented*) The method of claim 21, wherein (a) comprises mapping a plurality of 16-bit instructions to a plurality of PIWF configurations.

23. (*Previously Presented*) The method of claim 21, wherein (a) comprises mapping a plurality of 32-bit instructions to a plurality of PIWF configurations.

24. (*Previously Presented*) The method of claim 21, wherein (a) comprises mapping an instruction from a fill buffer to a PIWF configuration.

25. (*Previously Presented*) The method of claim 21, wherein (a) comprises mapping a 16-bit instruction to a PIWF configuration that has more than 32-bits.

26. (*Previously Presented*) The method of claim 21, wherein (a) comprises mapping a 32-bit instruction to a PIWF configuration that has more than 32-bits.

27. (*Previously Presented*) The method of claim 21, wherein (a) comprises performing a first partial mapping in parallel with (b), and performing a second partial mapping after (c).

28. (*Currently amended*) A cache controller for use with a processor, comprising:

a mapper for mapping each of a plurality of instructions to a predetermined instruction width format (PIWF) configuration;

a multiplexor for receiving said PIWF configurations from said mapper and for selecting, based on a unique tag associated with each of said plurality of instructions, in response to a selector signal, a desired one of said PIWF configurations for decoding and execution by the processor; and

a decoder for decoding the desired one of said PIWF configurations for execution by the processor.

29. (*Previously presented*) The cache controller of claim 28, wherein said mapper maps an instruction from a fill buffer to a PIWF configuration.

30. (*Previously presented*) A computer readable storage medium comprising a microprocessor core embodied in software, said microprocessor core including a cache controller comprising:

a plurality of mappers for receiving instructions of an instruction set, each mapper for mapping an instruction of said instruction set to a predetermined instruction width format (PIWF) configuration, wherein said plurality of mappers include:

at least one first mapper for receiving instructions from a fill buffer, and

at least one second mapper for receiving instructions from an instruction cache;

a multiplexor for receiving said PIWF configurations from said plurality of mappers and selecting, in response to a selector signal, a desired one of said PIWF configurations for decoding and execution by said microprocessor core, wherein the tangible computer readable storage medium consisting of at least one of a semiconductor disk, a magnetic disk and an optical disk; and

a decoder for decoding the desired one of said PIWF configurations for execution by the processor.

31. (*Previously presented*) A pipelined computer system that supports two or more instruction sets, the computer system comprising:

a cache memory adapted to output a plurality of instructions, wherein each instruction is associated with one of the instruction sets;

a cache controller having a parallel mapper to map each of the plurality of instructions into a predetermined instruction width format;

a tag comparator to compare a tag associated with each instruction in the plurality of instructions with a tag associated with a sought after address concurrently with the mapper mapping the plurality of instructions into the predetermined instruction width format; and

a selection circuit to select one of the mapped instructions for decoding, wherein the parallel mapper is disposed in the pipelined computer system upstream from the selection circuit, and wherein the parallel mapping, tag comparison, and selection are completed in a single pipeline stage; and

a decoder for decoding the selected mapped instruction for execution by a processor.

32. (*Previously presented*) The pipelined computer system of claim 31, further comprising a fill buffer for holding an instruction retrieved upon a cache miss, the instruction in the fill buffer being associated with one of the instruction sets and having a tag associated therewith, wherein

the parallel mapper maps the instruction held in the fill buffer to the predetermined instruction width format,

the tag comparator compares the tag associated with the instruction in the fill buffer to the tag of the sought after address concurrently with the mapper mapping the instruction in the fill buffer into the predetermined instruction width format, and

the selection circuit selects one of the mapped instructions or instruction held in the fill buffer for decoding, wherein the parallel mapper is disposed in the pipelined computer system upstream from the selection circuit, and wherein the parallel mapping, tag comparison, and selection are completed in a single pipeline stage.